UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,517	12/31/2003	Michael G. Lisanke	SOM920030007US1	9240

55420 7590 03/19/2009

FLEIT, GIBBONS, GUTMAN, BONGINI & BIANCO P.L
551 NW 77TH STREET
SUITE 111
BOCA RATON, FL 33487

EXAMINER	
WANG, HARRIS C	

ART UNIT	PAPER NUMBER
2439	

NOTIFICATION DATE	DELIVERY MODE
03/19/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptoboca@fggbb.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/750,517
Filing Date: December 31, 2003
Appellant(s): LISANKE ET AL.

Jon A. Gibbons
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed January 15, 2009 appealing from the Office action mailed May 12, 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.th

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

OpenPGP standard (RFC 2440) 11-1998

IBM Ceritification Study Guide AIX V4.3 System Administration, First Edition, Section 2.4.1 Using the alog command 5-1999

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Circenis (US 20040054908) in view of the OpenPGP standard (RFC 2440) further in view of The IBM Certification Study Guide AIX V4.5 System Administration (1999) (hereafter referred to as IBM).

Regarding Claims 1 and 3,

Circenis teaches a system that allows analysis of software running in a tamper-resistant environment, the system comprising *("A tamper-evident data management system…includes an application for collecting usage or metrics data from the computer system" Abstract).:*

a processor which monitors at least one instance of software execution, wherein the one instance is identified and selected by an end-user to be monitored by the processor, wherein the end-user is a user that initiates execution of the software at a system associated with the end-user is a user that initiates execution of the software at a system associated with the end-user and wherein the processor creates a log entry

with at least one of a set of data is used to diagnose the software execution in response to the one instance being identified and selected to be monitored; *("Using the tamper-evident system 200 of Fig. 3, a sender is able to monitor and control application utilization by collecting data associated with the application, creating tamper-evident data records, and providing the tamper-evident data records" Paragraph [0037])*

an encryption system which encrypts the log entry for the at least one set of data *(Figure 4 teaches encrypting the log entry for at least one set of data, particularly **step 320** "Sign data entry with application private key", **step 325** "Encrypt with vendor public key" and **step 330** "Store in data log")*

The Examiner interprets the processor that monitors software execution, wherein the one instance is identified by an end-user to be monitored by the processor as the user executing software on a device wherein each instance of execution is to be monitored by the processor, and in response to the one instance being executed a log entry is created (See Paragraph [0019] where Circenis teaches the pay per use application collecting metrics data to a data log).

Circenis does not explicitly teach an encryption system which generates at least one symmetric key and encrypts the log entry for the at least one set of data using the symmetric key, wherein the encryption system encrypts the symmetric key using a public key associated with the encryption system, wherein the log file includes the symmetric key which has been encrypted with the public key.

PGP ("Pretty Good Privacy") is a program that provides cryptographic privacy and authentication, and was created by Phillip Zimmermann in 1991. The OpenPGP standard (1998) is cited, but any PGP product teaches the generic method of:

1. Creating a message

2. Generating a symmetric key to be used as a session key for the message

3. Encrypting the session key using each recipient's public key. These "encrypted session keys" start the message.

4. The sending PGP encrypts the message using the session key, which forms the remainder of the message.

5. The receiving PGP device decrypts the session key using the recipient's private key

6. The receiving PGP decrypts the message using the session key.

Because Circenis already teaches one method of encrypting the data log, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the public-private key encryption of Circenis with the well known method of PGP, where the symmetric key is generated, the log entry is encrypted using the symmetric key, a public key encrypts the symmetric key, and the log file includes the symmetric key which has been encrypted with the public key.

The motivation is that PGP provides a more secure way of encrypting the log entries.

Circenis and OpenPGP do not explicitly teach a

a log file of a relatively-fixed size which stores the log entry for the at least one set of data which have been encrypted;

IBM teaches

a log file of a relatively-fixed size which stores the log entry for the at least one set of data which have been encrypted; *("The alog command can maintain and manage logs. It reads standard input , writes to standard output, and copies the output into a fixed-size file. This file is treated as a circular log" Section 2.4.1)*

a system for wrapping around and filling the log file from a beginning when the log file has been filled, allowing the log file to remain at a substantially-constant size even after the log file has been filled with data and a new entry is received. *("If the file is full, new entries are written over the oldest existing entries" Section 2.4.1). It is inherent that a circular log will wrap around and fill the log file from a beginning when the log file has been filled.*

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the monitoring system of Circenis to store the encrypted log entries in a circular log as described by IBM.

The motivation is that a circular log is a well known way to store a log file, where the circular log is inherently of a fixed size. It is inherent that a circular log will contain at least a pointer which identifies the next storage location for a next log entry.

The combined references of Circenis and IBM do not explicitly teach where

random data in the log file when it is originally created and which is replaced by log entries so that a size of the log including log entries appears to be a substantially-constant size;

It would have been obvious to one of ordinary skill in the art at the time of the invention to insert random data into the log file when it is initially created.

The motivation is to initialize the circular log.

Regarding Claim 2,

The combined references of Circenis, OpenPGP and IBM teach a system including the elements of claim 1 wherein the system includes a transmission system for sending the log file, upon command, to a secure processing location away from the system in which the log file was created. *("The data log may also be transmitted to a remote system (comprising, for example, the validation computer 150) over a network connection" Paragraph [0043] of Circenis, Figure 3 shows the transmission of the log file 115 to the secure processing location away from the system 150 Circenis)*

Regarding Claim 4,

The combined references of Circenis, OpenPGP and IBM teach a system including the elements of claim 1 wherein the system includes a mechanism for obscuring a log entry which has been created. *(Figure 4 of Circenis teaches encrypting the log entry for at least one set of data, particularly* ***step 320*** *"Sign data entry with application private key",* ***step 325*** *"Encrypt with vendor public key" and* ***step 330*** *"Store in data log")*

Regarding Claim 5,

The combined references of Circenis, OpenPGP and IBM teach a system including the elements of claim 4, Circenis further teaches the mechanism for obscuring the activity for which a log entry is created includes a printing function for writing into the log file.

("The customer site that forbids electronic media leaving the site may require that the vendor print out any validated and decrypted data logs and bring the printout back to the vendor site for processing and billing." Paragraph [0034] *Circenis*) The Examiner interprets printing out the data logs as the printing function.

Regarding Claim 6,

The combined references of Circenis, OpenPGP and IBM teach a system including the elements of claim 2 wherein the system includes a mechanism for receiving an indication from a user that transmission is desired and transmits the log file in response to that indication. *("Fig. 5 is a flowchart illustrating steps in validating the data. The program starts (step 355) and the data log is copied to the validation computer through an intermediary device or medium (step 360)" Circenis) Before the data can be validated there must inherently be some indication for the log file to be transmitted.*

Regarding Claim 7,

The combined references of Circenis, OpenPGP and IBM teach a system including the elements of claim 1 wherein the system further includes a mechanism for receiving an input from an end-user that initiates logging of log entries into the log file each time logging is desired by the user. *(Paragraph [0019] teaches a mechanism for receiving an input from an end-user that initiates logging of log entries, as shown by the "pay per use metering application that collects metrics data associated with computer system)* The user can use the system as many times as desired. The logger logs each use of the system. Therefore logging is initiated as desired by the end user.

Regarding Claim 8,

The combined references of Circenis, OpenPGP and IBM a system including the elements of claim 1 wherein the system further includes an initializing mechanism for determining each instance logging is to begin and initiating logging of log entries only in response to that initializing mechanism. *("The iCOD computer could save usage data to a log file or a central metering device" Paragraph [0024] Circenis) ("an iCOD computer residing on an isolated site should be designed to discourage any reverse engineering or other tampering and to make such tampering evident to the iCOD computer vendor" Paragraph [0023] Circenis) The Examiner interprets the iCOD inherently having an initializing mechanism. The Examiner interprets the design to discourage tampering as so that only logging entries are only initiated in response to the initializing mechanism.*

Regarding Claim 9,

The combined references of Circenis OpenPGP, and IBM teach a system including the elements of claim 1 wherein the system uses a public key to encrypt the log entry which has been created and a private key corresponding to the public key is used to decrypt the log which has been created at a secure location. *("Public and private encryption/decryption key pairs where data encrypted by a public key can only be decrypted with a corresponding private key, and visa versa, provide data confidentiality" Paragraph [0025], Figure 4 of Circenis shows encryption and Figure 5 shows decryption)*

Regarding Claim 10

Circenis teaches a method for diagnosing software in a tamper-resistant environment comprising the steps of:

monitoring at least one software operation activity within the tamper-resistant environment and generating messages in response to at least one instance of software execution within the tamper-resistant environment, wherein the software operation activity is identified and selected by an end-user to be monitored, wherein the end-user is a user that initiates execution of the software at a system associated with the end user; *("Using the tamper-evident system 200 of Fig. 3, a sender is able to monitor and control application utilization by collecting data associated with the application, creating tamper-evident data records, and providing the tamper-evident data records" Paragraph [0037])*

logging at least one software operation activity relating to a generated message by replacing a random data with an encrypted record of the software operation activity;

*(Figure 4 teaches encrypting the log entry for at least one set of data, particularly **step 320** "Sign data entry with application private key", **step 325** "Encrypt with vendor public key" and **step 330** "Store in data log")*

and sending the log file to a secure location where it the log file can be decrypted and analyzed; *("The data log may also be transmitted to a remote system (comprising, for example, the validation computer 150) over a network connection" Paragraph [0043] of Circenis, Figure 3 shows the transmission of the log file 115 to the secure processing location away from the system 150)*

and analyzing the decrypted log file data and providing information on the operation of the software in the tamper-resistant environment. *("The use of the vendor public and private keys ensures that only the vendor can decrypt the data logon the computer system…to preserve the confidentiality of the data log" Paragraph [0043]) It is inherent that the data log will provide information on the operation of the software in the tamper-resistant environment.*

The Examiner interprets the processor that monitors software execution, wherein the one instance is identified by an end-user to be monitored by the processor as the user executing software on a device wherein each instance of execution is to be monitored by the processor, and in response to the one instance being executed a log entry is created (See Paragraph [0019] where Circenis teaches the pay per use application collecting metrics data to a data log).

Circenis does not explicitly teach an encryption system which generates at least one symmetric key and encrypts the log entry for the at least one set of data using the symmetric key, wherein the encryption system encrypts the symmetric key using a public key associated with the encryption system, wherein the log file includes the symmetric key which has been encrypted with the public key.

PGP ("Pretty Good Privacy") is a program that provides cryptographic privacy and authentication, and was created by Phillip Zimmermann in 1991. The OpenPGP standard (1998) is cited, but any PGP product teaches the generic method of:

1. Creating a message
2. Generating a symmetric key to be used as a session key for the message
3. Encrypting the session key using each recipient's public key. These "encrypted session keys" start the message.
4. The sending PGP encrypts the message using the session key, which forms the remainder of the message.
5. The receiving PGP device decrypts the session key using the recipient's private key
6. The receiving PGP decrypts the message using the session key.

Because Circenis already teaches one method of encrypting the data log, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the public-private key encryption of Circenis with the well known method of PGP, where the symmetric key is generated, the log entry is encrypted using the symmetric key, a public

key encrypts the symmetric key, and the log file includes the symmetric key which has been encrypted with the public key.

The motivation is that PGP provides a more secure way of encrypting the log entries.

Circenis and OpenPGP do not explicitly teach

turning on logging and establishing a pointer for a location of a next logged software operation activity;

moving the pointer when a log entry has been made to a next available log position;

wrapping the pointer to a beginning of the log file when the log file is full of log entries;

IBM teaches turning on logging and establishing a pointer for a location of a next logged software operation activity; moving the pointer when a log entry has been made to a next available log position; *("The alog command can maintain and manage logs. It reads standard input, writes to standard output, and copies the output into a fixed-size file. This file is treated as a circular log" Section 2.4.1) It is inherent that a circular log has a pointer that moves to the next logged software operation activity.*

wrapping the pointer to a beginning of the log file when the log file is full of log entries; *("If the file is full, new entries are written over the oldest existing entries" Section 2.4.1). It is inherent that a circular log will wrap around and fill the log file from a beginning when the log file has been filled.*

The combined references of Circenis and IBM do not further teach generating a log file full of random data;

It would have been obvious to one of ordinary skill in the art at the time of the invention to insert random data into the log file when it is initially created.

The motivation is that it is inherent that the circular log is of a fixed size so it must be initialized with some values. One of ordinary skill in the art would know to initialize the circular log with random values.

Regarding Claim 11,

Circenis, OpenPGP and IBM teach a method including the steps of claim 10 wherein the step of turning on logging includes the steps of receiving an user input that logging is desired and initiating the logging in response thereto. *("The iCOD computer could save usage data to a log file or a central metering device that a vendor employee could check periodically by visiting the site." Paragraph [0024] Circenis) The Examiner interprets the vendor employee as the user the indicates logging is desired)*

Regarding Claim 12,

Circenis, OpenPGP and IBM teach a method including the steps of claim 10 wherein the step of at least one software operation activity further includes the steps of determining whether the software operation activity is to be logged, *The Examiner*

interprets that before the data is logged, inherently, there must be a step of determining whether the activity is to be logged.

and if so, determining when to encrypt the software operation activity to obscure what is being logged. *("Encryption may be added to keep the customer's data log confidential" Paragraph [0039] Circenis) The Examiner interprets that before the data log is encrypted there must inherently be a determining step of when to encrypt the software activity.*

Regarding Claim 13,

Circenis, OpenPGP and IBM teach a method including the steps of claim 10 wherein the step of logging the software operation activity further includes the steps of determining a next available log position, *It is inherent that a circular log requires determing a next available log position.*

replacing existing data in the location with the data from the software operation activity, *("If the file is full, new entries are written over the oldest existing entries" Section 2.4.1, IBM).*

and updating the pointer to provide a location of the next logged software operation activity. *It is inherent that a circular log updates the pointer to provide a location of the next activity.*

Regarding Claim 14,

Circenis, OpenPGP and IBM teach a method including the steps of claim 10 and further including the step of receiving a command from a user that indicates that sending the log file to a remote location is desired and transmitting the log file in response thereto. *("Fig. 5 is a flowchart illustrating steps in validating the data. The program starts (step 355) and the data log is copied to the validation computer through an intermediary device or medium (step 360)" Circenis) Before the data can be validated there must inherently be some indication for the log file to be transmitted.*

Regarding Claim 15,

Circenis teaches a method of analyzing the operation of software in a remote protected processing environment, the method including:

receiving from the remote protected processing environment an encrypted log file comprising at least one log entry with at least one set of data derived from at least one instance of software execution monitored in response to a user identifying and selecting the one instance of software execution, wherein the end-user is a user that initates execution of the software at a system associated with the end-user, whereby the set of data is used to diagnose the software execution; *("The data log also may be transmitted to a remote system (comprising, for example, the validation computer) over a network connection" Paragraph [0043])*

determining a decrypting key for the encrypted log file and decrypting the encrypted log file; *("The software on the validation computer may then decrypt each of the data log entries in the data log using the vendor private key" Paragraph [0043])*

analyzing the log entry at the remote protected processing environment and to determine whether an operation of the remote protected processing environment corresponding to the at least one set of data derived from at least one instance of software execution is appropriate; *("The data log is then further inspected by the vendor for evidence of customer tampering." Paragraph [0044])*

and reporting the results of the analyzing step. *(The Examiner interprets the vendor inspecting the data logs as reporting the results of the analyzing step)*

The Examiner interprets the processor that monitors software execution, wherein the one instance is identified by an end-user to be monitored by the processor as the user executing software on a device wherein each instance of execution is to be monitored by the processor, and in response to the one instance being executed a log entry is created (See Paragraph [0019] where Circenis teaches the pay per use application collecting metrics data to a data log).

Circenis does not explicitly teach an encryption system which generates at least one symmetric key and encrypts the log entry for the at least one set of data using the symmetric key, wherein the encryption system encrypts the symmetric key using a public key associated with the encryption system, wherein the log file includes the symmetric key which has been encrypted with the public key, determining a private decrypting key corresponding to the public key associated with the system, and using the decrypting key and the private decrypting key.

PGP ("Pretty Good Privacy") is a program that provides cryptographic privacy and authentication, and was created by Phillip Zimmermann in 1991. The OpenPGP standard (1998) is cited, but any PGP product teaches the generic method of:

1. Creating a message
2. Generating a symmetric key to be used as a session key for the message
3. Encrypting the session key using each recipient's public key. These "encrypted session keys" start the message.
4. The sending PGP encrypts the message using the session key, which forms the remainder of the message.
5. The receiving PGP device decrypts the session key using the recipient's private key
6. The receiving PGP decrypts the message using the session key.

Because Circenis already teaches one method of encrypting the data log, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the public-private key encryption of Circenis with the well known method of PGP, where the symmetric key is generated, the log entry is encrypted using the symmetric key, a public key encrypts the symmetric key, and the log file includes the symmetric key which has been encrypted with the public key.

The motivation is that PGP provides a more secure way of encrypting the log entries.

Circenis and OpenPGP does not explicitly teach that the data log is of substantially-constant size

IBM teaches that the data log is of substantially-constant size.
("The alog command can maintain and manage logs. It reads standard input , writes to standard output, and copies the output into a fixed-size file. This file is treated as a circular log" Section 2.4.1)

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the data log monitoring system of Circenis with the fixed-sized log (circular log) of IBM.

The motivation is that the circular log is well known in the art and without much modification the circular log can be used in the system of Circenis with no difference in result.

Regarding Claim 16,

Circenis, OpenPGP and IBM teach a method providing the steps of claim 15. It is inherent that before "the data log…may be transmitted to a remote system" (Paragraph [0043] *Circenis*) that an instruction to send the encrypted log file to the remote location is needed.

Circenis teaches including providing an instruction to initiate a logging of messages each time logging is desired by the user *("The iCOD computer could save usage data to a log file or a central metering device that a vendor employee could check periodically by visiting the site." Paragraph [0024]) The Examiner interprets the vendor employee as the user the indicates logging is desired)*

Regarding Claim 17,

Circenis, OpenPGP and IBM teach a method providing the steps of claim 16.

Circenis, OpenPGP and IBM do not explicitly teach wherein the instruction to initiate logging of messages includes the step of initiating programming within the remote protected processing environment to replace information in the encrypted log file with encrypted information relating to the operation of the remote protected processing environment.

It would have been obvious to one of ordinary skill in the art at the time of the invention to include programming within the remote system to replace information in the encrypted file log with encrypted information relating to the operation of the remote protected system.

The motivation is that in the system of Circenis, once the data log is passed to the remote system, it is in the hands of the vendor or system administrator. Because tampering is no longer an issue the vendor can adjust the data log to include whatever instruction is deemed necessary. One of ordinary skill in the art would be able to

replace encrypted data log information with encrypted information relating to the operation of the remote protected system.

Regarding Claim 18,

Circenis, OpenPGP and IBM teach a method providing the steps of claim 17.

Circenins and IBM do not explicitly teach wherein the step of replacing information in the encrypted log file includes the step of replacing random data which was placed in the encrypted log file when it was created.

It would have been obvious to one of ordinary skill in the art at the time of the invention to insert random data into the log file when it is initially created.

The motivation is that it is inherent that the circular log is of a fixed size so it must be initialized with some values. One of ordinary skill in the art would know to initialize the circular log with random values.

Regarding Claim 19,

Circenis, OpenPGP and IBM teach a method providing the steps of claim 17. IBM teaches a circular log wherein the step of replacing information in the log file inherently includes the step of using a pointer to a next location in the log file and the pointer wraps to a beginning the log file after the encrypted log file has been filled.

Regarding Claim 20,

Circenis teaches a computer program product for analyzing software running in a tamper-resistant environment, the computer program product comprising instructions for:

at least one set of data serviced from at least one instance of software execution identified and selected by an end-user to be monitored whereby the set of data is used to diagnose the software execution wherein the end-user is a user that initates execution of the software at a system associated with the end-user; *("Using the tamper-evident system 200 of Fig. 3, a sender is able to monitor and control application utilization by collecting data associated with the application, creating tamper-evident data records, and providing the tamper-evident data records" Paragraph [0037])*

encrypting the recording of the at least one set of data using a key; *(Figure 4 teaches encrypting the log entry for at least one set of data, particularly **step 320** "Sign data entry with application private key", **step 325** "Encrypt with vendor public key" and **step 330** "Store in data log")*

responding to a command and sending the encrypted log file comprising the at least one set of data which has been encrypted and sequentially recoded in the storage block to a remote location for decryption and analysis. *("The data log may also be transmitted to a remote system (comprising, for example, the validation computer 150) over a network connection" Paragraph [0043] of Circenis, Figure 3 shows the transmission of the log*

file 115 to the secure processing location away from the system 150). The Examiner interprets the data in the log of Circenis as being sequentially recoded. ("The sequence numbers of the data log entries are also checked for gaps or data log entries that are out of sequence" Paragraph [0044])

The Examiner interprets the processor that monitors software execution, wherein the one instance is identified by an end-user to be monitored by the processor as the user executing software on a device wherein each instance of execution is to be monitored by the processor, and in response to the one instance being executed a log entry is created (See Paragraph [0019] where Circenis teaches the pay per use application collecting metrics data to a data log).

Circenis does not explicitly teach an encryption system which generates at least one symmetric key and encrypts the log entry for the at least one set of data using the symmetric key, wherein the encryption system encrypts the symmetric key using a public key associated with the encryption system, wherein the log file includes the symmetric key which has been encrypted with the public key.

PGP ("Pretty Good Privacy") is a program that provides cryptographic privacy and authentication, and was created by Phillip Zimmermann in 1991. The OpenPGP standard (1998) is cited, but any PGP product teaches the generic method of:

1. Creating a message

2. Generating a symmetric key to be used as a session key for the message

3. Encrypting the session key using each recipient's public key. These "encrypted session keys" start the message.

4. The sending PGP encrypts the message using the session key, which forms the remainder of the message.

5. The receiving PGP device decrypts the session key using the recipient's private key

6. The receiving PGP decrypts the message using the session key.

Because Circenis already teaches one method of encrypting the data log, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the public-private key encryption of Circenis with the well known method of PGP, where the symmetric key is generated, the log entry is encrypted using the symmetric key, a public key encrypts the symmetric key, and the log file includes the symmetric key which has been encrypted with the public key.

The motivation is that PGP provides a more secure way of encrypting the log entries.

Circenis and OpenPGP do not explicitly teach

recording at least one set of data, which has been encrypted sequentially in a storage block of a substantially fixed size;

maintaining a pointer to a next available location for recording the at least one set of data sequentially in the storage block;

IBM teaches recording at least one set of data, which has been encrypted in a storage block of a substantially fixed size; *("The alog command can maintain and manage logs. It reads standard input , writes to standard output, and copies the output into a fixed-size file. This file is treated as a circular log" Section 2.4.1)*

It is inherent that a circular log maintains a pointer to a next available location for recording the at least one set of data sequentially in the storage block;

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the monitoring system of Circenis to store the encrypted log entries in a circular log as described by IBM.

The motivation is that a circular log is a well known way to store a log file, where the circular log is inherently of a fixed size. It is inherent that a circular log will contain at least a pointer which identifies the next storage location for a next log entry.

Regarding Claim 21,

Circenis, OpenPGP and IBM teach the computer program product of claim 20. Circenis and IBM do not further teach instructions for:

Initializing the storage block of a substantially fixed size with random information which has been encrypted to provide a block of apparent data.

It would have been obvious to one of ordinary skill in the art at the time of the invention to insert random data into the log file when it is initially created.

The motivation is that it is inherent that the circular log is of a fixed size so it must be initialized with some values. One of ordinary skill in the art would know to initialize the circular log with random values.

Regarding Claim 22,

Circenis, OpenPGP and IBM he computer program product of claim 20, further comprising instructions for:

writing the at least one set of data which has been encrypted and recorded events in a sequential order in the storage block *("The sequence numbers of the data log entries are also checked for gaps or data log entries that are out of sequence…Inconsistencies in…the sequence numbers would provide evidence of tampering with the data log" Paragraph [0044] of Circenis). Because the data log is supposed to be sequential, the Examiner interprets that the data is written in a sequential order.*

In a circular log it is inherent for wrapping around when an end of the storage block of the substantially fixed-size memory is reached.

(10) Response to Argument

The Appellant has provided evidence to support the new limitations added, so the 112 rejections are withdrawn.

The Appellant argues "As expressly taught by Circenis, the sender is the data owner, whereas a user in the presently claimed invention is an end user or an IT professional. (pg. 13)."

The Examiner respectfully disagrees with the Appellants interpretation of Circenis. If the data owner is using the system to monitor, then the data owner can be considered an "end user."

The Appellant then argues "The data owner of Circenis configures an application to monitor every instance of an application use…The presently claimed invention, on the other hand, is only monitoring a specific instance of a software execution.(pg. 14)"

However the claim language recites the limitation "a processor which monitors at least one instance of software execution" not "a specific instance of software execution" as the Appellant is arguing.

Furthermore, even if choosing to monitor every instance at least encompasses monitoring a specific instance. The claim limitation " a processor which monitors at least one instance of software execution , wherein the one instance is identified and selected" does not preclude the identification and selection of other instances.

The Appellant argues that "it is not customary to insert random data into log files….fixed size files do not have to be initialized with random data (pg. 16)."

The Examiner agrees that the "combined references of Circenis and IBM do not explicitly teach where random data in the log file when it is originally created (pg. 7 of

Final Office Action)." However the Examiner maintains “it would have been obvious to one of ordinary skill in the art at the time of the invention to insert random data into a log file."

Placing random data into a file and is recognized as part of the ordinary capabilities of one skilled in the art. Therefore the Examiner considers the limitation obvious because placing random data in a file would result in predictable results (i.e. data placed in a file)

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner’s answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Harris C Wang/

Examiner, Art Unit 2439

Conferees:

/Andrew L Nalven/

Primary Examiner, Art Unit 2434

/Kambiz Zand/

Supervisory Patent Examiner, Art Unit 2434